一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
COMP2014-6 (2014-04)

オンラインフレーム転送量最大化問題における競合比の改良

小林　浩二[†]　川原　純[††]　宮崎　修一[†††]

† 国立情報学研究所　〒 101–8430 東京都千代田区一ツ橋 2–1–2
†† 奈良先端科学技術大学院大学　情報科学研究科　〒 630–0192 奈良県生駒市高山町 8916–5
††† 京都大学　学術情報メディアセンター　〒 606–8501 京都市左京区吉田本町
E-mail: †kobaya@nii.ac.jp, ††jkawahara@is.naist.jp, †††shuichi@media.kyoto-u.ac.jp

あらまし　本稿では，k 個のパケットからなるフレーム転送量最大化問題（k-FTM）と呼ばれる，ネットワーク上におけるスイッチのオンラインバッファ管理問題の一問題について考える．本問題は，大きなフレームが k 個のパケットに分割されてインターネット上で転送され，受信者は k 個全てのパケットを受信できたときに限りフレームを再構成可能という状況をモデル化したものである．Kesselman らは本問題に対して，$k=2$ の場合でさえ任意のアルゴリズムの競合比は発散することを示した．そこで，入力に制限を加えた k-FTM（k-OFTM）を考え，任意のバッファの大きさ $B(\geqq k)$ に対して，その競合比が高々 $\frac{2kB}{\lfloor B/k \rfloor}+k$ となるアルゴリズムを考案した．また，彼らは $2B \geqq k$ かつ k が 2 の冪のとき，任意の決定性アルゴリズムの競合比の下限 $\frac{B}{\lfloor 2B/k \rfloor}=\Omega(k)$ を示した．本稿において，我々は k-OFTM に対する競合比の上限と下限を改良している．主要な結果として，彼らの上界 $O(k^2)$ を $\frac{5B+\lfloor B/k \rfloor-4}{\lfloor \lfloor B/k \rfloor /2 \rfloor}=O(k)$ へ改良し，下界 $\Omega(k)$ に漸近的に一致させている．また，任意の $k \geqq 2$ と任意の B に対する，決定性アルゴリズムの競合比の下限 $2k-1$ と，任意の $k \geqq 3$ と任意の B に対する，確率アルゴリズムの初めての非自明な競合比の下限 $k-1$ を与える．
キーワード　オンライン問題，競合比解析，バッファ管理，パケット断片化

Improved bounds for online k-frame throughput maximization in network switches

Koji M. KOBAYASHI[†], Jun KAWAHARA[††], and Shuichi MIYAZAKI[†††]

† National Institute of Informatics　2–1–2 Hitotsubashi, Chiyoda-ku, Tokyo, 101–8430, Japan
†† Graduate School of Information Science, Nara Institute of Science and Technology
8916–5 Takayama, Ikoma, Nara, 630–0192, Japan
††† Academic Center for Computing and Media Studies, Kyoto University
Yoshida-Honmachi, Sakyo-ku, Kyoto, 606–8501, Japan
E-mail: †kobaya@nii.ac.jp, ††jkawahara@is.naist.jp, †††shuichi@media.kyoto-u.ac.jp

Abstract　We consider a variant of the online buffer management problem in network switches, called the k-frame throughput maximization problem (k-FTM). This problem models the situation where a large frame is fragmented into k packets and transmitted through the Internet, and the receiver can reconstruct the frame only if he/she accepts all the k packets. Kesselman et al. introduced this problem and showed that its competitive ratio is unbounded even when $k=2$. They also introduced a restricted variant of k-FTM, called k-OFTM. They proposed an online algorithm and showed that its competitive ratio is at most $\frac{2kB}{\lfloor B/k \rfloor}+k$ for any $B \geq k$, where B is the size of the buffer. They also presented a lower bound of $\frac{B}{\lfloor 2B/k \rfloor}=\Omega(k)$ for deterministic online algorithms when $2B \geqq k$ and k is a power of 2. In this paper, we improve upper and lower bounds on the competitive ratio of k-OFTM. Our main result is to improve an upper bound of $O(k^2)$ by Kesselman et al. to $\frac{5B+\lfloor B/k \rfloor-4}{\lfloor B/2k \rfloor}=O(k)$ for $B \geqq 2k$, which matches their lower bound of $\Omega(k)$ asymptotically. We also present an improved lower bound of $\frac{2B}{\lfloor B/(k-1) \rfloor}+1$ on the competitive ratio of deterministic online algorithms for any $k \geqq 2$ and any $B \geqq k-1$, and the first nontrivial lower bound of $k-1$ on the competitive ratio of randomized algorithms for any $k \geqq 3$ and any B.
Key words　Online Problem, Competitive Analysis, Buffer Management, Packet Fragmentation

1. ま え が き

データをインターネットを介して転送するとき，各データはより小さな断片に分割され，そのような断片はデータパケットに包まれる．パケットはネットワーク上の幾つかのスイッチやルータを経由し，受信者に転送され，受信側で元のフレームが再構成される．高い転送量の達成を妨げるボトルネックの 1 つはスイッチやルータの処理能力である．もし単位時間に到着するパケット数が，単位時間にスイッチが処理出来るパケット数を超えるなら，幾つかのパケットは廃棄されなければならない．この不便さを緩和するため，スイッチは後で処理されるパケットを一時的に格納する FIFO バッファを備えるのが通常である．バッファ管理のポリシーはネットワーク全体の性能に影響するため重要である．

Aiello ら [1] は**競合比解析** [10], [32] を用いてバッファ管理問題の解析を初めて行った．問題の入力はイベントの列によって構成される．各イベントは，到着イベントか送信イベントである．到着イベントでは，バッファ（FIFO キュー）の入力ポートにパケットが 1 つ到着する．パケットは単位長であり，その価値はパケットの優先度を示す．バッファは同時に高々B 個のパケットを蓄えることが出来る．到着イベントの際，バッファが満杯であるなら，新しく到着したパケットは非受理となる．バッファに空きがあるならば，アルゴリズムは受理するかどうかをオンラインに判断することになる．各送信イベントには，FIFO キューの先頭からパケットを 1 つ送信する．アルゴリズムの利得は送信されるパケットの価値の総和であり，本問題の目的は，それの最大化である．競合比解析においては，オンラインアルゴリズムの得た利得を，入力が全て分かっている場合の，最適なオフラインアルゴリズム（これを以下では OPT と書く）の得る利得と比較する．任意の入力に対して，オンラインアルゴリズムの得る利得が，OPT の得る利得の $1/c$ 倍以上である場合に，そのオンラインアルゴリズムの競合比は c であると言う．

Aiello ら [1] の研究の後に，バッファ管理問題の競合比解析について多数の研究が行われた．例えば，Andelman ら [5] は [1] の 2 値モデルを，パケットの優先度が任意の値を取る多値モデルへと拡張した．その他の拡張の 1 つは**プリエンプション**の許可である．すなわち，アルゴリズムはバッファの中に存在するパケットを廃棄出来る．これらのモデルの競合比の結果は [3], [4], [12], [18], [20], [33] で与えられている．また，1 つのキューに限らず，複数のキューのスイッチ [2], [5]〜[7], [9], [28]，共有メモリ型スイッチ [14], [19], [27]，CIOQ スイッチ [8], [21], [22], [25]，クロスバースイッチ [23], [24] 等，様々なモデルに対するバッファ管理も広範囲に研究されている．網羅的な調査論文は [13] を参照せよ．

Kesselman ら [26] はバッファ管理問題のモデルのもう 1 つの自然な拡張として，最初に述べた筋書き，すなわち受信したパケットからのフレームの再構成を動機とする，k フレーム転送量最大化問題（k-FTM）を提案した．本問題では，データの塊を**フレーム**と呼び，フレームが k 個のパケットに分割され，インターネットを介して転送される．($j \in [1,k]$ について，フレームの j 番目のパケットを j-パケットと呼ぶ.）受信側では，あるフレームを構成する k 個のパケットの全てを受信すれば，そのフレームを再構成出来る．(このときフレームは**完成された**と言う.）それ以外のとき，すなわち k 個のパケットのうち 1 個でも失うと，受信側は何も得られない．本問題の目標は完成されたフレームの最大化である．Kesselman ら [26] はこのフレームの再構成の問題を 1 つの FIFO キューに対して考えた．彼らは最初に k-FTM の任意の決定性アルゴリズムの競合比は $k=2$ の場合でさえ発散することを示した．(議論を少し修正すると，確率アルゴリズムに対しても同様のことを示せる.）しかし，この下限の証明のために用いられた入力は，現実世界の状況からかけ離れている．すなわち，TCP/IP ネットワークのようなネットワークにおいて，各パケットは一般に出発の順に到着するが，証明で用いられた入力では，フレーム f_i の 1-パケットはフレーム $f_{i'}$ の 1-パケットより前に到着し，一方で f_i の 2-パケットは $f_{i'}$ の 2-パケットより後に到着する．これを踏まえて，彼らは入力列に対してより自然な設定を導入し，これを**順序遵守** (order-respecting) と呼んだ．概略を言うと，2 つのフレーム f_i と $f_{i'}$ のそれぞれの j-パケットの到着順は f_i と $f_{i'}$ のそれぞれの j'-パケット ($j' < j$) の到着順に従う．(正式な定義は 2 節で与えられる.）このように入力が制限された問題を**順序遵守 k フレーム転送量最大化問題**（k-OFTM）と呼ぶ．彼らは $2B \geq k$ かつ k が 2 の冪であるとき，k-OFTM の決定性アルゴリズムの競合比の下限 $B/\lfloor 2B/k \rfloor = \Omega(k)$ を示した．彼らはプリエンプション無しの StaticPartitioning (SP) と呼ばれる決定性アルゴリズムを設計し，任意の $B \geq k$ に対して，k-OFTM の決定性アルゴリズムの競合比の上限 $\frac{2kB}{\lfloor B/k \rfloor} + k = O(k^2)$ を示した．彼らは真の競合比は $\Theta(k)$ であると予想している．

1.1 我々の結果

本稿では，k-OFTM について以下の結果を示す．

(i) 任意の $k \geqq 2$ と任意の $B \geqq 2k$ に対して，決定性アルゴリズム Middle-Drop and Flush (MF) を設計し，その競合比が高々 $\frac{5B+\lfloor B/k \rfloor -4}{\lfloor B/2k \rfloor}$ であることを示す．この比は $O(k)$ であり，これは Kesselman ら [26] の上界 $O(k^2)$ を改良し，下界 $\Omega(k)$ と漸近的に一致する．それ故，我々は彼らの予想を肯定的に解決したことになる．

(ii) 任意の $k \geqq 2$ と任意の $B \geqq k-1$ に対して，決定性アルゴリズムの競合比の下限 $\frac{2B}{\lfloor B/(k-1) \rfloor} + 1$ を与える．これは，以前の下限 $\frac{B}{\lfloor 2B/k \rfloor}$ を約 4 倍改善する．さらに，任意の $B \leq k-2$ に対して，任意の決定性アルゴリズムの競合比は発散することを示す．

(iii) 任意の $k \geqq 3$ と任意の B に対して，オブリビアスな入力に対する確率アルゴリズムの競合比の下限 $k-1$ を示す．これは初めて得られた非自明な下限であり，(i) で述べた決定性アルゴリズムの上限と漸近的に一致する．これは本問題で確率を用いたアルゴリズムを設計しても，確率を用いない場合と比べて競合比を大きく改善することはできないことを意味する．

1.2 使用する技法

我々のアルゴリズム MF の着想を簡潔に説明する．Kesselman ら [26] のアルゴリズム SP は以下のように動作する．(1) バッファを仮想的に均等に k 個のサブバッファに分割する．各サブバッファ（$j \in [1,k]$ について j-**サブバッファ**と呼ぶ）の大きさは $A = \lfloor \frac{B}{k} \rfloor$ である．j-サブバッファは j-パケットのみを格納するために用いられる．(2) もし，j-サブバッファが溢れたら，すなわち，j-サブバッファに j-パケットが A 個格納されているときに j-パケットが到着したら，その新しく到着した j-パケットを非受理する（「末尾廃棄」ポリシー）．SP は多数の j-パケットがバースト的に到着するとき，得られる利得が小さくなり，これにより SP の競合比は $\Omega(k^2)$ に悪くなる．

本稿では，より良いアルゴリズムを設計するために 2 つの主要な着想を導入する．1 つは，「末尾廃棄」ポリシーの代わりに「**中央廃棄**」ポリシーを採用する．このポリシーでは，新しく j-パケットが到着し，j-サブバッファが満杯であるとき，j-サブバッファの $\lfloor A/2 \rfloor + 1$ 番目のパケットをプリエンプションし，到着した j-パケットを受理する．もう 1 つは，フラッシュと呼ばれる操作で，ある瞬間にある条件を満たす全てのパケットをプリエンプションする．2 つの着想は，以下で説明されるように，競合比を $O(k)$ に改良する上で本質的である．MF は与えられる全フレームの集合を，1-パケットの到着順を基にした規則を用いて，**ブロック** $BL_1, BL_2, \ldots$ に分割する．（規則は 3.1 節の MF の定義で説明される．ここで，ブロック BL_i は，後の節で述べる**ブロック番号** i のフレームの集合に一致する．）それぞれのブロックは $3B$ 個のフレームを含む．各ブロックは**良い**ブロックと**悪い**ブロックに分類される．入力の最初は全てのブロックは良いブロックである．ある瞬間に，（ブロック BL_i に含まれるパケットが非受理またはプリエンプションされることによって）BL_i に含まれる少なくとも $\lfloor A/2 \rfloor$ 個のフレームが完成される可能性が無くなったとき，BL_i は悪いブロックになる．そのような場合，MF は BL_i に含まれるフレームの完成を完全に諦め，BL_i に属するフレームのパケットがバッファに存在するならば，それらを全てプリエンプションする．（この操作をフラッシュと呼ぶ．）入力の終了時に MF は良いブロック 1 つあたり，少なくとも $\lfloor A/2 \rfloor$ 個のフレームを完成させていることに注意せよ．

ブロック BL_i が良いブロックから悪いブロックに変わる瞬間を考えよう．これは，（ある j について）BL_i のあるフレームの j-パケット p が（j-サブバッファから）プリエンプションされたときのみ起こる．中央廃棄ポリシーの性質より，以下の性質を満たす 2 つの整数 i_1 と i_2 ($i_1 < i < i_2$) の存在を示すことが出来る．(i) フラッシュ操作の直後，BL_{i_1} と BL_{i_2} は良いブロックであり，$BL_{i_1+1}, BL_{i_1+2}, \ldots, BL_{i_2-1}$ の全ては悪いブロックである．(ii) フラッシュ操作の直前，BL_{i_1} と BL_{i_2} にそれぞれ属する j-パケット p_1 と p_2 がバッファの中に存在する．(iii) フラッシュ操作の直前，完成される可能性のあるフレームに属する BL_i の全ての j-パケット（p を含む）は p_1 と p_2 の間に位置する．上記の (ii) は i_2 が i_1 より遥かに大きな場合（それ故，BL_{i_1} と BL_{i_2} に多数のブロックがある場合）でさえ p_1 と p_2 の到着時刻は近いことを意味する（何故なら，p_2 が到着した時，p_1 はバッファ内にまだ存在するため）．BL_{i_1} から BL_{i_2} までのブロックに属するフレームの j-パケットが短い時間間隔でバースト的に到着し，それ故，任意のアルゴリズムは（最適オフラインアルゴリズム OPT でさえも）それらの多くを受理できない．この方法で，OPT によって受理される，2 つの連続する良いブロックの間のブロックに属するフレームのパケットの数（すなわち，2 つの連続する良いブロックの間のブロックに属する OPT によって完成されるフレームの数）を抑えることが出来る．より正確には，もし BL_{i_1} と BL_{i_2} が入力の終了時に連続する 2 つの良いブロックであるなら，$BL_{i_1}, BL_{i_1+1}, \ldots, BL_{i_2-1}$ に含まれる OPT によって完成されたフレームの数が高々$5B + A - 4 = O(B)$ 個であることを (i) を用いて示すことが出来る．BL_{i_1} は良いブロックであるので，MF は BL_{i_1} に含まれる $\lfloor A/2 \rfloor = \Omega(B/k)$ 個のフレームを完成させることに注意せよ．これにより MF の競合比は $O(k)$ となる．

1.3 関連研究

上で述べた結果に加えて，Kesselman ら [26] は任意の B について，$k \geqq 3$ のとき，プリエンプション可能な k-OFTM の貪欲アルゴリズムの競合比が発散することを示した．彼らは k-FTM のオフライン版も考え，近似アルゴリズムに関する幾つかの結果を示した．最近，川原と小林 [16] によって，2-OFTM の最適な競合比が 3 であることが示された．これは貪欲アルゴリズムによって達成されている．

Scalosub ら [31] は**フレームグッドプット最大化**問題と呼ばれる k-FTM を一般化した問題を提案した．この問題では，フレームの集合は**ストリーム**を構成し，同じストリームのパケットの到着順に制約が課せられている．彼らは競合比が $O((kMB+M)^{k+1})$ の決定性アルゴリズムを設計した．ここで，M はストリームの数である．さらに彼らは任意の決定性アルゴリズムの競合比の下界 $\Omega(kM/B)$ を示した．

Emek ら [11] は**オンライン集合充填問題**を提案し，k-FTM との関連を指摘した．この問題は，各フレームが異なる数（高々 $k_{\max}$ 個）のパケットを含むという点で，k-FTM とは異なる．また，フレーム f について，f は $s(f)$ 個のパケットから構成されるとすると，f に属する $s(f)(1-\beta)$ 個のパケットが転送されれば，f は再構成可能となる．ここで，β $(0 \leqq \beta < 1)$ は与えられたパラメータである．もう 1 つのパラメータとして，スイッチの容量 c が与えられる．到着イベントでは，キューの入力ポートに幾つかのパケットが到着する．スイッチはすぐにそれらのうち c 個を転送でき，残りのパケットに対し，バッファ管理アルゴリズムを適用する．すなわち，（もし存在すれば）残りのパケットの中からバッファに格納するパケットを決める．Emek らは Priority と呼ばれる確率アルゴリズムを設計し，$\beta = 0$ かつ $B = 0$ であるとき，競合比が $k_{\max}\sqrt{\sigma_{\max}}$ であることを示した．ここで，$\sigma_{\max}$ は同時に到着するパケットの最大数である．彼らは任意の確率アルゴリズムの下限 $k_{\max}\sqrt{\sigma_{\max}}(\log\log k/\log k)^2$ も与えている．Mansour ら [29] は，全てのフレームがそれぞれちょうど k 個のパケットを含むとき，任意の β について，Priority の競合比が $8k\sqrt{\sigma_{\max}(1-\beta)/c}$ であることを示した．さらに，幾つかの一問題が研究されている [15], [30]．

2. 問題設定

本節では，順序を遵守する k 個のパケットからなるオンラインフレーム転送量最大化問題（k-OFTM）の正式な定義を与える．**フレーム** は k 個のパケット $p_1, \ldots, p_k$ からなる．バッファ（FIFO キュー）が 1 つ存在し，同時に高々B 個のパケットを蓄えることが出来る．入力は 0 番目から始まるフェイズの列である．i 番目のフェイズは，**到着サブフェイズ**と，それに続く**送信サブフェイズ**からなる．到着サブフェイズでは，幾つかのパケットがバッファに到着する．アルゴリズムの役割は到着する各パケット p に対して，p を**受理する**，もしくは p を**非受理する**か決定することである．また，アルゴリズムはバッファに空きを作るために，現在のバッファ内に存在するパケット p' を捨てることが出来る．（この場合，アルゴリズムは p' を**プリエンプションする**と言う．）もしパケット p が非受理されるかプリエンプションされるなら，p は廃棄されると言う．受理されたパケットは，キューの最後尾に蓄えられる．同じサブフェイズに到着して受理されたパケットは，任意の順序でキューに入れることが出来る．送信サブフェイズには，バッファが空でなければ，キューの先頭のパケットが転送される．技術的な理由のため，必ずパケットは 1 つ以上到着する入力のみを考える．

パケット p が i 番目の到着フェイズに到着したとき，$\mathrm{arr}(p) = i$ と書く．$\mathrm{arr}(p_1) \leqq \cdots \leqq \mathrm{arr}(p_k)$ を満たす任意のフレーム $f = \{p_1, \ldots, p_k\}$ について，p_i を f の **i-パケット**と呼ぶ．$\mathrm{arr}(p_{i,1}) \leqq \cdots \leqq \mathrm{arr}(p_{i,k})$ と $\mathrm{arr}(p_{i',1}) \leqq \cdots \leqq \mathrm{arr}(p_{i',k})$ を満たす 2 つのフレーム $f_i = \{p_{i,1}, \ldots, p_{i,k}\}$ と $f_{i'} = \{p_{i',1}, \ldots, p_{i',k}\}$ について考える．もし，任意の整数 j と j' について，$\mathrm{arr}(p_{i,j}) \leqq \mathrm{arr}(p_{i',j})$ ならば，またそのときに限り $\mathrm{arr}(p_{i,j'}) \leqq \mathrm{arr}(p_{i',j'})$ であるなら，f_i と $f_{i'}$ は**順序遵守**であると言う．もし入力列 σ の任意の 2 つのフレームが順序遵守であるなら，σ は**順序遵守**であると言う．任意のアルゴリズム ALG，ALG の任意のフレーム $f = \{p_1, \ldots, p_k\}$ に対して，f を構成する全てのパケット，すなわち，$p_1, \ldots, p_k$ が ALG によって転送された場合，ALG の f は**完成された**と言う．k-FTM の目的は，完成されたフレームの数の最大化である．k-OFTM は，順序遵守である入力に制限された k-FTM である．

アルゴリズム ALG の**利得**は，完成されたフレームの数であり，それを $V_{ALG}(\sigma)$ と表記する．もし ALG が確率アルゴリズムならば，ALG の利得は期待値 $\mathbb{E}[V_{ALG}(\sigma)]$ として定義される．ここで，期待値は ALG の確率的な選択について取られる．OPT を最適なオフラインアルゴリズムとする．任意の入力 σ に対して，$V_{ALG}(\sigma) \geqq V_{OPT}(\sigma)/c$ (ALG が確率アルゴリズムの場合，$\mathbb{E}[V_{ALG}(\sigma)] \geqq V_{OPT}(\sigma)/c$) が成立するとき，$A$ の**競合比**は c であると言う．一般性を失わないので，OPT は決してプリエンプションを行わないと仮定する．また，OPT は決して完成されたフレーム以外のフレームを構成するパケットを受理しないと仮定する．

3. 上　　限

本節では，我々が考案したアルゴリズム Middle-Drop and Flush (MF) の定義を与え，その競合比の解析を行う．

3.1 アルゴリズム

最初に MF の記述に必要な記号の定義を与える．n 個のパケット $p_1, p_2, \ldots, p_n$ が i 番目の到着サブフェイズに，MF のバッファに到着すると仮定する．各パケットについて，MF は 1 つずつ（以下で定義する順番で）それを受理するか非受理するかを決める．t_{p_j} を MF がパケット p_j を処理する時刻とする．t_{p_j} を p_j の**決定時刻**と呼ぶ．従って，$p_1, p_2, \ldots, p_n$ がこの順に処理されるなら，$t_{p_1} < t_{p_2} < \cdots < t_{p_n}$ が成り立つ．（便宜上，後の解析では，OPT も p_j を t_{p_j} と同じ時刻に処理をする．）また，i 番目の送信サブフェイズでバッファの先頭からパケットを転送する時刻を i 番目の送信サブフェイズの**送信時刻**と呼ぶ．決定時刻または送信時刻のことを**イベント時刻**と呼び，それ以外の時刻を**非イベント時刻**と呼ぶ．非イベント時刻の間は，バッファの配置は変化しないことに注意せよ．任意のイベント時刻 t に対して，t とその次のイベント時刻の間の任意の瞬間を $t+$ で表す．同様に，t とその前のイベント時刻の間の任意の瞬間を $t-$ で表す．

ALG を MF または OPT とする．非イベント時刻 t，フレーム f のパケット p について，ALG が時刻 t より前に f の任意のパケットが廃棄されていないとき，すなわち，f が t の時点で完成され得るフレームであるとき，p は時刻 t に ALG にとって**有効**であると言う．この場合，f は時刻 t に ALG にとって**有効**であるとも言う．完成されたフレームは入力の終了時に有効である．j-パケット p と非イベント時刻 t について，もし p が時刻 t に MF のバッファに格納されているなら，$\ell(t, p)$ を「1+(p の前に置かれている j-パケットの数)」と定義する．すなわち，p は MF のバッファの $\ell(t, p)$ 番目の j-パケットである．もし p が時刻 t にまだ到着していないなら，$\ell(t, p) = \infty$ と定義する．

MF の実行中，MF は以下の貪欲アルゴリズム GR_1 を同じ入力に対して仮想的に走らせる．概略を言えば，GR_1 は 1-パケットに対してのみ貪欲で，$j(\in [2, k])$-パケットは無視する．正式に述べると，GR_1 は同じ大きさ B の FIFO キューを用いる．p の到着時に，もし p が $j(\in [2, k])$-パケットであるなら，GR_1 はそれを非受理する．もし p が 1-パケットならば，GR_1 はキューに空きがあるならそれを受理する．送信サブフェイズでは，通常通り GR_1 はキューの先頭のパケットを転送する．

MF は 2 つの内部変数 `Counter` と `Block` を用いる．`Counter` は GR_1 の受理したパケットの数を $3B$ で割った数を数えるために用いる．`Block` は正の整数の値を取る．最初は 1 で，`Counter` が 0 にリセットされるたびに 1 ずつ増加する．

$A = \lfloor B/k \rfloor$ と定義する．MF は各 j について，高々A 個の j-パケットをバッファに格納する．$j = 1$ のとき，MF は GR_1 の振る舞いを以下の方法で参照する．2 つの内部変数 `Counter` と `Block` を用い，MF は GR_1 によって受理された 1-パケットを，到着順に従ってブロックに分割する．それぞれのブロックは $3B$ 個の 1-パケットを含む．MF は各ブロックの最初の A 個の 1-パケットを受理し，残りを非受理する．$j \geqq 2$ のとき，MF は有効でない j-パケットを無視する．有効な j-パケット

p を処理するとき，もし MF は既にキューの中に A 個の j-パケットを格納しているなら，MF は「中央」の j-パケットをプリエンプションし，p を受理する．

非イベント時刻 t について，$b(t)$ を時刻 t における Block の値とする．パケット p について，p のブロック番号 $g(p)$ を以下で定義する．1-パケット p について，$g(p) = b(t-)$ とする．ここで t は p の決定時刻である．$j(\in [2,k])$-パケット p について，$g(p) = g(p')$ とする．ここで，p' は p と同じフレームに属する 1-パケットである．それ故，同じフレームに属する全てのパケットは同じブロック番号である．また，フレームのブロック番号も自然な方法で定義する．すなわち，フレーム f のブロック番号 $g(f)$ は f を構成するパケットの（一意に定まる）ブロック番号である．非イベント時刻 t と正の整数 u について，$h_{ALG,u}(t)$ を ALG のフレーム f が時刻 t に有効で，$g(f) = u$ を満たすフレーム f の数とする．

到着サブフェイズでは，2 つ以上のパケットがキューに到着するかもしれないことを思い出そう．MF はパケットをフレームのパケット順（すなわち，フレーム内で何番目のパケットか），そしてブロックの番号の順に処理する．もしそれらの両方が等しければ，任意の順に処理する．言い換えると，MF はそれらのパケットを以下の規則で処理する．i-パケット p と i'-パケット p' を考える．もし $i < i'$ ならば，p は p' より前に処理をし，$i' < i$ ならば，p' は p より前に処理をする．もし $i = i'$ であり，$g(p) < g(p')$ ならば，p は p' より前に処理をし，$i = i'$ であり，$g(p') < g(p)$ ならば，p' は p より前に処理をする．もし $i = i'$ であり，$g(p) = g(p')$ ならば，処理順は任意である．正式な MF の記述は以下の通りである．

Middle-Drop and Flush Algorithm

初期化: Counter $:= 0$, Block $:= 1$.

j-パケット p が到着したとする．

Case 1: $j = 1$，すなわち，p が 1-パケットの場合:

Case 1.1: GR_1 **が** p **を非受理する場合:**

MF は p を非受理する．

Case 1.2: GR_1 **が** p **を受理する場合:**

Counter $:=$ Counter $+1$.

Case 1.2.1: Counter $\leqq A$ **の場合:**

MF は p を受理する．（MF は必ず p を受理出来る．補題 4(c) を参照．）

Case 1.2.2: $A <$ Counter $< 3B$ **の場合:**

MF は p を非受理する．

Case 1.2.3: Counter $= 3B$ **の場合:**

MF は p を非受理する．

Counter $:= 0$ とし，Block $:=$ Block $+ 1$ とする．

Case 2: $j \geqq 2$，すなわち，p が 1-パケットでない場合:

Case 2.1: 時刻 t_p- に p が有効でない場合:

MF は p を非受理する．

Case 2.2: 時刻 t_p- に p が有効である場合:

Case 2.2.1: 時刻 t_p- にバッファ内の j-パケットの数が高々 $A-1$ である場合:

MF は p を受理する．

Case 2.2.2: そうでない場合，すなわち，時刻 t_p- にバッファ内の j-パケットの数が（少なくとも）A である場合:

t_p- の MF のバッファ内の先頭から $\lfloor A/2 \rfloor + 1$ 番目の j-パケット，すなわち，$\ell(t_p-, p') = \lfloor A/2 \rfloor + 1$ が成立する p'，をプリエンプションし，p を受理する．また，t_p- におけるバッファ内に p' と同じフレームのパケットが存在するなら，それらを全てプリエンプションする．

Case 2.2.2.1: $h_{MF,g(p')}(t_p-) \leqq \lfloor A/2 \rfloor$ **である場合:**

バッファ内の $g(p'') = g(p')$ が成立するパケット p'' を全てプリエンプションする.（これをフラッシュと言う.）

Case 2.2.2.2: $h_{MF,g(p')}(t_p-) \geqq \lfloor A/2 \rfloor + 1$ **である場合:**

何もしない．

3.2 解析の概要

入力が終了した後の任意の時刻を τ とする．c を τ における Counter の値とする．$c = 0$ ならば，$M = b(\tau) - 1$ と定義し，$c > 0$ ならば，$M = b(\tau)$ と定義する．任意のフレーム f について，$1 \leqq g(f) \leqq M$ であることに注意せよ．整数の集合 G を $G = \{M\} \cup \{i \mid MF$ は $g(f) = i$ を満たすフレーム f を少なくとも $\lfloor A/2 \rfloor$ 個完成させる $\}$ と定義する．$m = |G|$ とする．各 $j \in [1, m]$ について，a_j を G の中の j 番目に小さい数とする．もしあるブロック番号が G に含まれるなら，それを良いブロック番号と呼び，そうでなければ，悪いブロック番号と呼ぶ．a_j は j 番目の良いブロック番号であり，特に，$M \in G$ であるので $a_m = M$ であることに注意せよ．我々の最初の要の補題は以下である．

［補題 1］ $a_1 = 1$ が成り立つ．

入力の終了時に有効な任意のフレームは完成されているので，$V_{OPT}(\sigma) = \sum_{i=1}^{M} h_{OPT,i}(\tau)$ が得られ，$V_{MF}(\sigma) = \sum_{i=1}^{M} h_{MF,i}(\tau) \geqq \sum_{i=1}^{m} h_{MF,a_i}(\tau)$ が得られる．

最初に，良いブロック番号のブロックのフレームに含まれる MF が完成させたフレーム数を評価する．以下は G の定義から言える．

任意の $i \in [1, m-1]$ について，$h_{MF,a_i}(\tau) \geqq \lfloor A/2 \rfloor$ が成り立つ． (1)

次に m 番目の良いブロック番号 M に注目する．それは他のブロック番号とは異なる性質を持つので，別に議論をする．

［補題 2］ (a) もし $c = 0$ または $c \in [\lfloor A/2 \rfloor, 3B-1]$ が成り立つならば，$h_{MF,M}(\tau) \geqq \lfloor A/2 \rfloor$ が成り立つ．(b) もし $c \in [1, \lfloor A/2 \rfloor - 1]$ かつ $M \geqq 2$ が成り立つならば，$h_{MF,M}(\tau) + B - 1 \geqq h_{OPT,M}(\tau)$ が成り立つ．(c) もし $c \in [1, \lfloor A/2 \rfloor - 1]$ かつ $M = 1$ が成り立つならば，$h_{MF,M}(\tau) \geqq h_{OPT,M}(\tau)$ が成り立つ．

また，良いブロック番号の観点から，OPT の完成したフレームの数を評価する．

［補題 3］ (a) $h_{OPT,M}(\tau) \leqq 4B - 1$ が成り立つ．(b) $\sum_{j=a_1}^{a_2-1} h_{OPT,j}(\tau) \leqq 4B + A - 3$ が成り立つ．(c) 任意の

$i \in [2, m-1]$ について，$\sum_{j=a_i}^{a_{i+1}-1} h_{OPT,j}(\tau) \leqq 5B + A - 4$ が成り立つ．

以上の不等式を用いることで，M と c の値による場合分けによって MF の競合比が得られる．初めに，少なくとも 1 つのパケットが到着すると仮定しているので，$V_{OPT}(\sigma) > 0$ であり，$M = 1$ が成り立つならば $c \geqq 1$ である．今，もし $M = 1$ で $c \in [1, \lfloor A/2 \rfloor - 1]$ ならば補題 2(c) より $h_{MF,1}(\tau) \geqq h_{OPT,1}(\tau)$ が成り立つ．$h_{MF,1}(\tau) \geqq h_{OPT,1}(\tau) = V_{OPT}(\sigma) > 0$ であるから，$\frac{V_{OPT}(\sigma)}{V_{MF}(\sigma)} = \frac{h_{OPT,1}(\tau)}{h_{MF,1}(\tau)} \leqq 1$ が成り立つ．もし，$M = 1$ で $c \in [\lfloor A/2 \rfloor, 3B - 1]$ ならば，補題 2(a) と補題 3(a) から $\frac{V_{OPT}(\sigma)}{V_{MF}(\sigma)} = \frac{h_{OPT,1}(\tau)}{h_{MF,1}(\tau)} \leqq \frac{4B-1}{\lfloor A/2 \rfloor} < \frac{5B+A-4}{\lfloor A/2 \rfloor}$ が成り立つ．

もし，$M \geqq 2$ で $c \in \{0\} \cup [\lfloor A/2 \rfloor, 3B-1]$ ならば，補題 3 から $V_{OPT}(\sigma) = \sum_{i=1}^{M} h_{OPT,i}(\tau) = \sum_{i=1}^{m-1} \sum_{j=a_i}^{a_{i+1}-1} h_{OPT,j}(\tau) + h_{OPT,a_m}(\tau) \leqq (m-1)(5B + A - 4) - B + 1 + (4B - 1) < m(5B + A - 4)$ が成り立つ.（補題 1 より $a_1 = 1$ が成り立ち，また $a_m = M$ が成り立つことに注意せよ.）また，(1) と補題 2(a) から $V_{MF}(\sigma) \geqq \sum_{i=1}^{m} h_{MF,a_i}(\tau) \geqq m\lfloor A/2 \rfloor$ が成り立つ．従って，$\frac{V_{OPT}(\sigma)}{V_{MF}(\sigma)} < \frac{5B+A-4}{\lfloor A/2 \rfloor}$ が成り立つ．最後に，もし $M \geqq 2$ で $c \in [1, \lfloor A/2 \rfloor - 1]$ ならば，補題 2(b) と補題 3(b)，(c) より $V_{OPT}(\sigma) = \sum_{i=1}^{M} h_{OPT,i}(\tau) = \sum_{i=1}^{m-1} \sum_{j=a_i}^{a_{i+1}-1} h_{OPT,j}(\tau) + h_{OPT,a_m}(\tau) \leqq (m-1)(5B + A - 4) - B + 1 + h_{OPT,M}(\tau) \leqq (m-1)(5B + A - 4) + h_{MF,M}(\tau)$ が成り立つ．また，(1) より $V_{MF}(\sigma) = \sum_{i=1}^{m} h_{MF,a_i}(\tau) \geqq (m-1)\lfloor A/2 \rfloor + h_{MF,M}(\tau)$ が成り立つ．従って，$\frac{V_{OPT}(\sigma)}{V_{MF}(\sigma)} \leqq \frac{(m-1)(5B+A-4)+h_{MF,M}(\tau)}{(m-1)\lfloor A/2 \rfloor + h_{MF,M}(\tau)} < \frac{5B+A-4}{\lfloor A/2 \rfloor}$ が成立する．

以上より全ての場合で $\frac{V_{OPT}(\sigma)}{V_{MF}(\sigma)} < \frac{5B+A-4}{\lfloor A/2 \rfloor}$ を証明した．$\frac{5B+A-4}{\lfloor A/2 \rfloor} = \frac{5B+\lfloor B/k \rfloor-4}{\lfloor B/2k \rfloor}$ に注意することで以下の定理を得る．

［定理 1］ $B/k \geqq 2$ のとき，MF の競合比は高々 $\frac{5B+\lfloor B/k \rfloor-4}{\lfloor B/2k \rfloor}$ である．

3.3 解　　析

本稿では，MF の実行可能性のみを示し，補題 1，2，3 の証明は省略する．省略された証明は本稿の完全版 [17] を参照せよ．

［補題 4］ 任意の正の整数 z に対して，GR_1 が $z(\geqq 2B)$ 個の 1-パケットを受理するとし，GR_1 が i 番目に受理するパケットを p_i $(i \in [1, z])$ で表す．このとき，以下が成立する．(a) MF が p_j を受理する様な任意の $j(\in [1, z - 2B + 1])$ に対して，$\mathrm{del}_{MF}(p_j) < \mathrm{arr}(p_{j+2B-1})$ が成立する．(b) 任意の $u(\geqq 0)$ に対して，MF はパケット集合 $\{p_{3Bu+1}, \ldots, p_{3Bu+A}\}$ のパケットを受理することが出来る．また，それらのブロック番号は，$u + 1$ である．(c) 1-パケット p の受理を判断する直前に，$0 \leqq$ `Counter` $\leqq A - 1$ が成立している場合，MF のバッファに，その 1-パケット p を受理するための空きが存在する．

証明 (a) 補題の条件を満たす 1-パケット p_j と非イベント時刻 $t_{p_j}+$ を考えよ．$t_{p_j}+$ は，GR_1 と MF が p_j を受理する瞬間であることに注意せよ．MF はその定義より，受理した 1-パケットを必ず転送する．また，バッファの大きさは B であり，各サブフェイズに転送可能なパケットは 1 つだけであるので，MF は B フェイズ内に p_j を転送する．すなわち，$t_{p_j}+$ から p_j を転送する前の間の，送信サブフェイズの数は，高々 $B - 1$ である．結果として，GR_1 は，この期間に，高々 $B - 1$ 個のパケットを転送出来る．一方，GR_1 も p_j を受理するので，$t_{p_j}+$ における GR_1 のバッファ内のパケット数は少なくとも 1 個である．すなわち，$t_{p_j}+$ における GR_1 のバッファ内の空きは，高々 $B - 1$ である．以上より，t_{p_j+} から MF が p_j を転送する非イベント時刻の間に，GR_1 が受理可能なパケットの数は，高々 $(B - 1) + (B - 1) = 2B - 2$ である．つまり，GR_1 が p_{j+2B-1} をバッファに入れる前に，MF は p_j を転送する．よって，$\mathrm{del}_{MF}(p_j) < \mathrm{arr}(p_{j+2B-1})$ が成立する．

(b) まず，MF は各 $x(\in [1, k])$-パケットを A 個まで受理可能であることに注意せよ．Case 1.2.1, 1.2.2, 1.2.3 より，MF はパケット $p_1, p_2, \cdots, p_A$ を受理し，$p_{A+1}, p_{A+2}, \cdots, p_{3B}$ を非受理する．この間，`Block` は 1 であり，MF が p_{3B} を非受理した直後，`Block` は 2 になり，`Counter` は 0 となる．(a) の証明より，$\mathrm{del}_{MF}(p_A) < \mathrm{arr}(p_{A+2B-1})$ が成立するので，$\mathrm{del}_{MF}(p_A) < \mathrm{arr}(p_{3B+1})$ が成立する．すなわち，$t_{p_{3B+1}}-$ の時点で MF のバッファ内には 1-パケットは 1 つも存在しない．ゆえに，MF は，パケット $p_{3B+1}, p_{3B+2}, \ldots, p_{3B+A}$ を受理する．よって，帰納的に，(b) を示すことが出来る．

(c) 各 $u = 0, 1, \ldots$ に対して，$p_{3Bu+1}, \ldots, p_{3Bu+A}$ の決定時刻の直前に，$0 \leqq$ `Counter` $\leqq A - 1$ が成立している．よって，(b) の証明より，(c) が成立する．

□

4. 決定性アルゴリズムの下限

本節では，決定性アルゴリズムに対する下限を示す．

［定理 2］ $k \geqq 2$ とする．$B \geqq k - 1$ ならば，任意の決定性アルゴリズムに対する競合比は少なくとも $\frac{2B}{\lfloor B/(k-1) \rfloor} + 1$ である．また，$B \leqq k - 2$ ならば，任意の決定性アルゴリズムに対する競合比は発散する．

証明 オンラインアルゴリズム ALG を固定して考える．次の様な入力 σ を考える．0 番目のフェイズに $2B$ 個の 1-パケットが到着する．このとき，ALG は高々 B 個のパケットを受理する．一方で，OPT は ALG が受理しないパケットを B 個受理する．ALG が受理するパケットの集合を C と呼び，OPT が受理するパケットの集合を D と呼ぶ．$i(\in [1, B-1])$ 番目のフェイズには，パケットは到着しない．従って，$B - 1$ 番目のフェイズの直後,（B 回の送信サブフェイズが起こるので）ALG と OPT の両方のキューが空になる．

B 番目のフェイズで，最初の $2B$ 個の 1-パケットと同様に，$B + \lfloor \frac{B}{k-1} \rfloor$ 個の 1-パケットが到着する．このとき，ALG は高々 B 個のパケットを受理する．一方で，OPT は ALG が受理しないパケットを $\lfloor \frac{B}{k-1} \rfloor$ 個だけ受理する．ALG（OPT）が受理するパケットの集合を E（F）と呼ぶ．$i(\in [B+1, 2B-1])$ 番目のフェイズには，パケットは到着しない．

$2B$ 番目のフェイズに，もう一度同様に $2B$ 個の 1-パケットが到着する．このとき，ALG は高々 B 個のパケットを受理する．一方で，OPT は ALG が受理しないパケットを B 個受理する．ALG（OPT）が受理するパケットの集合を G（H）と

呼ぶ．以上で，入力中の全ての 1-パケットは到着した．

各 $j = 2, \ldots, k$ に対して，$3B + (j-2)B = (j+1)B$ 番目のフェイズに，パケット集合 D の B 個の j-パケットが到着し，OPT はそれらを全て受理し転送する．(ALG はそれらを受理する意味は無い．) 次に，$(k+2)B$ 番目のフェイズに，パケット集合 $C \cup E \cup F \cup G$ の 2-パケットから k-パケット全てが一度に到着する．このとき，ALG が受理出来るそれらのパケットは高々 B 個であるので，ALG が完成出来るフレームの数は高々 $\lfloor \frac{B}{k-1} \rfloor$ 個である．一方，OPT は，F に対応する $\lfloor \frac{B}{k-1} \rfloor (k-1)$ 個のパケットを受理する．これは $\lfloor \frac{B}{k-1} \rfloor (k-1) \leqq B$ が成り立つので可能であることに注意せよ．従って，OPT は，F に対応する $\lfloor \frac{B}{k-1} \rfloor$ 個のフレームを全て完成させる．

その到着フェイズの後，パケット集合 H の 2-パケットから k-パケットが到着し，OPT はそれらを全て受理し転送する．入力は順序遵守であることに注意せよ．

以上の議論より，$V_{ALG}(\sigma) \leqq \lfloor \frac{B}{k-1} \rfloor$ と $V_{OPT}(\sigma) = 2B + \lfloor \frac{B}{k-1} \rfloor$ が成立する．よって，$B \geqq k-1$ ならば，$\frac{V_{OPT}(\sigma)}{V_{ALG}(\sigma)} \geqq \frac{2B + \lfloor \frac{B}{k-1} \rfloor}{\lfloor \frac{B}{k-1} \rfloor} = \frac{2B}{\lfloor \frac{B}{k-1} \rfloor} + 1$ が成立する．また，$B \leqq k-2$ ならば，競合比は発散する．

□

5. 確率アルゴリズムの下限

本節では，確率アルゴリズムに対する下限を示す．

［定理 3］ $k \geqq 3$ に対して，任意の正の定数 ϵ について，オブリビアスな入力に対する任意の確率アルゴリズムの競合比は少なくとも $k - 1 - \epsilon$ である．

証明 任意の確率アルゴリズム ALG を固定して考える．y を後で固定する大きな整数とする．我々が構築する入力 σ は $(k-1)yB$ 個のフレームから構成される．これらのフレームはそれぞれ yB 個のフレームを含む，$k-1$ 個のグループに分割される．また，各グループは，それぞれ B 個のフレームを含む，y 個のサブグループからなる．各 $i(\in [1, k-1])$ と各 $j(\in [1, y])$ について，$F(i,j)$ を i 番目のグループの j 番目のサブグループに含まれるフレームの集合とし，$F(i) = \cup_j F(i,j)$ とする．各 $x(\in [1,k])$ について，$P(i,j,x)$ は $F(i,j)$ に含まれるフレームの x-パケットの集合とする．$P(i,x) = \cup_j P(i,j,x)$ とする．

最初に入力 σ の構築法の概略を単純化して述べる．上で定義した $k-1$ 個のグループのうち，後で述べる決め方で，ALG の振る舞いに応じて，1 つを良いグループと決める．入力の前半部（0 番目から $(k-1)yB - 1$ 番目のフェイズ）では，ALG が良いグループを見分けることができないような方法でパケットが到着する．また，バッファの大きさは抑えられているので，ALG は前半部の間，多くのフレームを諦めなければならない．ALG の yB 個のフレームのみが前半部の終わりに廃棄されずに残る．入力の後半部では，良いグループ以外のグループの k-パケットがバースト的に到着する．一方で，良いグループの k-パケットは 1 つずつ到着する．それ故，もしアルゴリズムが前半部の終わりに良いグループ（グループ 1 と呼ぶ）の多くのパケットを運良く確保したとき，結果として多くのフレームを完成させることが出来る．しかしながら，そのようなアルゴリズムはグループ 1 が良いグループではないような入力に対しては得られる利得が小さくなる．それ故，オンラインアルゴリズムの最も良い戦略は（それが確率アルゴリズムであっても），前半部で各グループから等しい数のフレームを確保することである．

入力 σ の構築法を示す前に，入力の部分列を定義する．任意の整数 t について，$P(i,j,x)$ に含まれる B 個のパケットが t 番目のフェイズに到着し，$t+1$ 番目から $(t+B-1)$ 番目のフェイズにはパケットが到着しないとする．このとき，この部分列を **t 番目のフェイズから始まる $P(i,j,x)$ のサブラウンド**と呼ぶ．1 つのサブラウンドに着目するとき，アルゴリズムはサブラウンドの終了時に $P(i,j,x)$ の全てのパケットを受理と転送出来ることに注意せよ．**t 番目のフェイズから始まる $P(i,x)$ のラウンド**は $(t + (j-1)B)$ 番目のフェイズから始まる $P(i,j,x)$ のサブラウンドを $j = 1, \ldots, y$ について連結したものである．

我々の入力の前半部は，全ての $i \in [1, k-1]$ と全ての $x \in [1, k-1]$ について，$(i + x - 2)yB$ 番目のフェイズから始まる $P(i,x)$ のラウンドによって構成される．任意の 2 つのラウンドについて，$i + x = i' + x'$ ならば $P(i,x)$ のラウンドと $P(i',x')$ のラウンドは同時に開始することに注意せよ．ここで，$x = k$ のときの $P(i,x)$ に含まれるパケットの到着についてはまだ特定していないことに注意せよ．この部分が我々の構築法の要であり，以下で簡単に説明する．

$(k-2)yB$ 番目のフェイズに始まる $k-1$ 個の $(P(1,k-1), P(2,k-2), \cdots, P(k-1,1)$ の$)$ ラウンドについて考えよう．これらのラウンドは同時に始まる．各 j について，これら $k-1$ 個のラウンドの j 番目のサブラウンドでは ALG はバッファの大きさの制約から，($(k-1)B$ 個のうち）高々 B 個のパケットだけ受理出来る．各 $j \in [1, y]$ について，$A_{i,j}$ を $P(i,j,k-i)$ に含まれる ALG が受理するパケットの数の期待値とする．上の議論から，$\Sigma_i A_{i,j} \leqq B$ が成り立ち，それ故，$\Sigma_i \Sigma_j A_{i,j} \leqq yB$ が成り立つ．$A_i = \Sigma_j A_{i,j}$ とし，A_z を $A_1, A_2, \cdots, A_{k-1}$ の中の最小値（複数ある場合は任意）とする．$\Sigma_i A_i = \Sigma_i \Sigma_j A_{i,j} \leqq yB$ であるので $A_z \leqq \frac{yB}{k-1}$ が成り立つことに注意せよ．また，A_i は期待値であるので，z は ALG の記述のみから決定される（ALG の実際の振る舞いには依存しない）ことに注意せよ．

今，$P(i,k)$ $(i \in [1, k-1])$ に含まれるパケットの到着について説明する．$i \neq z$ について，$P(i,k)$ に含まれる全ての yB 個のパケットが $(i + k - 2)yB$ 番目のフェイズに同時に到着する．$i = z$ については，通常の $(z+k-2)yB$ 番目から始まる $P(z,k)$ のラウンドと同じである．入力 σ が順序遵守であることを検証するのは難しくない．また，σ の構築法は ALG の実際の振る舞いには依存しないので，σ がオブリビアスであることを検証するのは簡単である．特に，z は $A_{i,j}$ $(i \in [1, k-1], j \in [1, y])$ の値のみに依存し，σ は ALG を実際に実行しながらではなく，実行前に構築可能である．

最初に，OPT は任意の x について $P(z,x)$ に含まれる全てのパケットを受理でき，転送出来ることに注意せよ．従って，OPT は $F(z)$ に含まれる yB 個のフレームを全て完成でき，

$V_{OPT}(\sigma) \geqq yB$ となる．一方，$P(i,k)$ $(i \neq z)$ に含まれる全てのパケットは同時に到着するので，ALG はそのうち高々 B 個のパケットを受理でき，それ故，各 $i(\neq z)$ について，$F(i)$ に含まれる高々 B 個のフレームを完成出来る．また，ALG は $F(z)$ に含まれる高々 $A_z \leqq \frac{yB}{k-1}$ 個のフレームを完成出来る．従って，$\mathbb{E}[V_{ALG}(\sigma)] \leqq \frac{yB}{k-1} + (k-2)B$ が成り立つ．もし，y を不等式 $y \geqq \frac{(k-1)^2(k-2)}{\epsilon} - (k-1)(k-2)$ を満たすように選ぶなら，$\frac{V_{OPT}(\sigma)}{\mathbb{E}[V_{ALG}(\sigma)]} \geqq \frac{yB}{(yB)/(k-1)+(k-2)B} = k-1-\frac{(k-1)^2(k-2)}{y+(k-1)(k-2)} \geqq k-1-\epsilon$ が得られる． □

謝辞 第二著者は，JSPS 科研費 23700001 の助成を受けた．第三著者は，JSPS 科研費 24500013 の助成を受けた．

文　献

[1] W. Aiello, Y. Mansour, S. Rajagopolan, and A. Rosén, "Competitive queue policies for differentiated services," *Journal of Algorithms*, Vol. 55, No. 2, pp. 113–141, 2005.
[2] S. Albers and M. Schmidt, "On the performance of greedy algorithms in packet buffering," *SIAM Journal on Computing*, Vol. 35, No. 2, pp. 278–304, 2005.
[3] N. Andelman, "Randomized queue management for DiffServ," *In Proc. of the 17th ACM Symposium on Parallel Algorithms and Architectures*, pp. 1–10, 2005.
[4] N. Andelman and Y. Mansour, "Competitive management of non-preemptive queues with multiple values," *In Proc. of the 17th International Symposium on Distributed Computing*, pp. 166–180, 2003.
[5] N. Andelman, Y. Mansour and A. Zhu, "Competitive queueing policies for QoS switches," *In Proc. of the 14th ACM-SIAM Symposium on Discrete Algorithms*, pp. 761–770, 2003.
[6] Y. Azar and A. Litichevskey, "Maximizing throughput in multi-queue switches," *Algorithmica*, Vol.45, No.1, pp, 69–90, 2006,
[7] Y. Azar and Y. Richter, "Management of multi-queue switches in QoS networks," Algorithmica, Vol.43, No.1-2, pp, 81–96, 2005.
[8] Y. Azar and Y. Richter, "An improved algorithm for CIOQ switches," *ACM Transactions on Algorithms*, Vol. 2, No. 2, pp. 282–295, 2006,
[9] M. Bienkowski, "An optimal lower bound for buffer management in multi-queue switches," *In Proc. of the 16th ACM-SIAM Symposium on Discrete Algorithms*, pp. 1295–1305, 2010.
[10] A. Borodin and R. El-Yaniv, "Online computation and competitive analysis," *Cambridge University Press*, 1998.
[11] Y. Emek, M. Halldórsson, Y. Mansour, B. Patt-Shamir, J. Radhakrishnan and D. Rawitz, "Online set packing and competitive scheduling of multi-part tasks," *In Proc. of the 29th ACM symposium on Principles of Distributed Computing*, pp. 440–449, 2010.
[12] M. Englert and M. Westermann, "Lower and upper bounds on FIFO buffer management in QoS switches," *Algorithmica*, Vol.53, No.4, pp, 523–548, 2009,
[13] M. Goldwasser, "A survey of buffer management policies for packet switches," ACM SIGACT News, Vol.41, No. 1, pp.100–128, 2010.
[14] E. Hahne, A. Kesselman and Y. Mansour, "Competitive buffer management for shared-memory switches," *In Proc. of the 13th ACM Symposium on Parallel Algorithms and Architectures*, pp. 53–58, 2001.
[15] M. M. Halldórsson, B. Patt-Shamir and D. Rawitz, "Online scheduling with interval conflicts," *In Proc. of the 28th Symposium on Theoretical Aspects of Computer Science*, pp. 472–483, 2011.
[16] J. Kawahara, and K. M. Kobayashi, "Optimal buffer management for 2-frame throughput maximization," *In Proc. of the 20th international colloquium on Structural Information and Communication Complexity*, pp. 274–285, 2013.
[17] J. Kawahara, K. M. Kobayashi and S. Miyazaki, "Better bounds for online k-frame throughput maximization in network switches," *arXiv*:1309.4919 [cs.DS], 2013.
[18] A. Kesselman, Z. Lotker, Y. Mansour, B. Patt-Shamir, B. Schieber, and M. Sviridenko, "Buffer overflow management in QoS switches," *SIAM Journal on Computing*, Vol. 33, No. 3, pp. 563–583, 2004.
[19] A. Kesselman and Y. Mansour, "Harmonic buffer management policy for shared memory switches," *Theoretical Computer Science*, Vol. 324, No.2-3, pp. 161–182, 2004.
[20] A. Kesselman, Y. Mansour and R. van Stee, "Improved competitive guarantees for QoS buffering," *Algorithmica*, Vol.43, No.1-2, pp. 63–80, 2005.
[21] A. Kesselman and A. Rosén, "Scheduling policies for CIOQ switches," *Journal of Algorithms*, Vol. 60, No. 1, pp. 60–83, 2006.
[22] A. Kesselman and A. Rosén, "Controlling CIOQ switches with priority queuing and in multistage interconnection networks," *Journal of Interconnection Networks*, Vol. 9, No. 1/2, pp. 53–72, 2008.
[23] A. Kesselman, K. Kogan and M. Segal, "Packet mode and QoS algorithms for buffered crossbar switches with FIFO queuing," *In Proc. of the 27th ACM symposium on Principles of Distributed Computing*, pp. 335–344, 2008.
[24] A. Kesselman, K. Kogan and M. Segal, "Best effort and priority queuing policies for buffered crossbar switches," *In Proc. of the 15th international colloquium on Structural Information and Communication Complexity*, pp. 170–184, 2008.
[25] A. Kesselman, K. Kogan and M. Segal, "Improved competitive performance bounds for CIOQ switches," Algorithmica, Vol.63, No.1-2, pp, 411–424, 2012.
[26] A. Kesselman, B. Patt-Shamir and G. Scalosub, "Competitive buffer management with packet dependencies," Theoretical Computer Science, Vol.489–490, pp. 75–87, 2013.
[27] K. Kobayashi, S. Miyazaki and Y. Okabe, "A tight bound on online buffer management for two-port shared-memory switches," *In Proc. of the 19th ACM Symposium on Parallel Algorithms and Architectures*, pp. 358–364, 2007.
[28] K. Kobayashi, S. Miyazaki and Y. Okabe, "Competitive buffer management for multi-queue switches in QoS networks using packet buffering algorithms," *In Proc. of the 21st ACM Symposium on Parallel Algorithms and Architectures*, pp. 328–336, 2009.
[29] Y. Mansour, B. Patt-Shamir, and D. Rawitz, "Overflow management with multipart packets," *In Proc. of the 31st IEEE Conference on Computer Communications*, pp. 2606–2614, 2011.
[30] Y. Mansour, B. Patt-Shamir, and D. Rawitz, "Competitive router scheduling with structured data," *In Proc. of the 9th Workshop on Approximation and Online Algorithms*, pp. 219–232, 2012.
[31] G. Scalosub, P. Marbach and J. Liebeherr, "Buffer management for aggregated streaming data with packet dependencies," *In Proc. of the 29th IEEE Conference on Computer Communications*, pp. 1–5, 2010.
[32] D. Sleator and R. Tarjan, "Amortized efficiency of list update and paging rules," *Communications of the ACM*, Vol. 28, No. 2, pp. 202–208, 1985.
[33] M. Sviridenko, "A lower bound for on-line algorithms in the FIFO model," unpublished manuscript, 2001.